P1394.1 Working Group
July 29 - 30, 1996
San Jose, CA

The meeting was called to order by Dick Scheel at 9:00 AM. As is customary, the participants signed in and introduced themselves. A list of attendees is appended to these minutes.

The draft agenda was presented for approval and the addition of new items:

1. Administrative matters
   1.1 Approval of minutes
   1.2 PAR status
   1.3 Bylaws
   1.4 Election of chair
2. Draft 0.01 P1394.1
3. Network simulator program
4. Design discussions
   4.1 Broadcast transactions
   4.2 Reset notification / asynchronous transaction integrity
   4.3 Heartbeat
   4.4 Isochronous routing
5. Meeting schedule

1. ADMINISTRATIVE MATTERS

1.1 Approval of minutes

The minutes of the last meeting, May 23, were approved by unanimous consent.

1.2 PAR

Dick Scheel distributed the PAR approved by the IEEE at the June 20 meeting of NesCom and noted the change in project number from the proposed P1394.3 to the actually assigned P1394.1. The IEEE noted additional coordination requested by others, IEC TC100 United States working group and ISO/IEC JTC1/SC26 United States Technical Advisory Group. The coordination is to be effected by joint distribution of working drafts and, at the option of the requesting group, attendance at P1394.1 working group meetings. Dick Scheel took an action item to determine the appropriate points of contact, to which the Technical Editor, Peter Johansson, will distribute working drafts in the future.

IEEE also requests a revision of the project's target date to something more realistic than the current December 31, 1996.

1.3 Bylaws

Bylaws, and the need for them, were discussed. Voting standards were suggested as the most important ground rule for the group. Dick suggested that the criteria be a) any attendee at a meeting who has attended two out of the last three meetings (including the current meeting) and b) all of the attendees of the first two meetings are qualified to vote. An informal motion to this effect passed unanimously.

Dick Scheel indicated that the preference of the as yet unelected Chair is to conduct the meetings informally, with the proviso that closer adherence to Robert's Rules of Order may be invoked at the Chair's discretion.

1.4 Elections

Dick Scheel was uncontestedly elected as the working group's Chair.

2. P1394.1 WORKING DRAFT 0.01

The revision was presented and accepted as a starting point for the meeting’s technical discussions. A PDF file will be made available on the working group’s FTP site as soon as possible. The editor reported some technical difficulties in the creation of a single PDF file from the FrameMaker 4 source document(s) and will report back as soon as these have been surmounted.

3. NETWORK SIMULATOR

Dick Scheel reported on a C++ simulation program under development by Sony that models the behavior of a Serial Bus with an arbitrary number of nodes. The goal is to add the working group’s definitions of bridges to this model and use it as a test bed for our ideas.

The model is driven by an input script file that can create the modeled environment and then apply events to the model. The output of the model is reported in a trace file.

4. DESIGN DISCUSSIONS

4.1 Reset notification / asynchronous transaction integrity

Peter Johansson led a discussion of the current (incompletely articulated) scheme for reset notification and the bridge manager mechanisms to provide for transaction integrity.

The crux of the issue appears to be the necessity to flush the outbound transaction queue for each node that is engaged in “conversation” with remote nodes. It may be desirable to impose some form of hierarchical structure upon this problem (the NODE_ENABLE register is one example), but the underlying nature remains the same.

After a productive morning discussion, the consensus was that the basic scheme is workable but that it requires a more detailed elucidation. Dave James, Peter Johansson and Dick Scheel resolved to hold one or more editing sessions to produce a description at a sufficient level of detail for the working group to review at the next meeting.

4.2 Heartbeat

After lunch, discussion resumed with the matter of a “heartbeat” between the bridge manager and the bridge portals.

The heartbeat could exist at a location in the bridge manager and not require any periodic writes to the bridge portals. Any bridge manager that wished to claim ownership of a bridge portal would first determine if the incumbent bridge manager is alive. If not, the bridge manager could safely usurp ownership of the portal.

The bridge manager could have a HEARTBEAT register which is defined to increase monotonically with a period of N. Dave James suggested a value of 10 milliseconds. Dick Scheel proposed that the period be expressed in terms of a minimum (10 milliseconds) and maximum (100 milliseconds) period and that the address of a block of memory (rather than a CSR) be provided as the location of a heartbeat.

The next topic was the necessity or lack thereof for a heartbeat mechanism by which a bridge lets the bridge manager know it is still in good health. If a portal experiences a command reset or a power reset, it is sufficient for the portal to reset its attached Serial Bus. If there is another portal connected to the Serial Bus, that portal presumably knows the location of the bridge manager and sends a RESET_NOTIFICATION message. Note that this places a burden on the bridge manager to perform a scan of the just reset bus to determine whether extant bridge portals have vanished or new bridge portals are apparent. The group was uncertain whether there would be cases where a bridge portal or bridge fabric (between portals) could “hang” without any notification to any external device. It was mentioned that some implementations might use a hardware watchdog timer to reset themselves in such a case, especially if their bridge implementation could have that type of failure.

4.3 Isochronous routing

After reintroducing the group to the concepts of channel remapping and the (potentially) multicast nature of isochronous channels as they traverse bridges, Dick Scheel suggested that a considerable simplification could be obtained if bridges performed no remapping of channel numbers. In this case, the bridge manager might reserve a number of channels whose extent was known to be global across the Serial Bus net. Then applications would petition the bridge manager for a universal channel number if they wished to talk across Serial Bus bridges. Peter Johansson questioned whether or not this was a significant functional increment to bridges in light of the fact that they must already examine the data stream in order transform the time stamps.

4.4 Remote transactions

Peter Johansson described the use of the REMOTE_DESTINATION, REMOTE_PAYLOAD and REMOTE_REQUEST registers to generate remote transactions. Dave James offered a modification to support portal relative addressing mode in the REMOTE_REQUEST register and further suggested that the control and status functions of REMOTE_REQUEST be split into two registers.

These suggestions eliminate the necessity for the “portal-specific” window described in the current draft and it will be deleted.

The minimum size of the REMOTE_PAYLOAD register needs to be increased to two octlets to accommodate lock transactions.

4.5 Broadcast

Dick Scheel opened a discussion on the necessity for broadcast transactions. Broadcast is advanced as a possibility to support the transport of IP, ARP and RARP over Serial Bus.

Dave Banks suggested that Serial Bus isochronous channels might be used to transport the ARP and RARP packets. The disadvantage of this approach is the bursty nature of the data; bandwidth would have to be allocated beforehand but would not be used in most cycles.

Other transport protocols carry IP traffic without broadcast capabilities, e.g., ATM. Couldn’t the same strategies be used for Serial Bus? Perhaps there could be an ARP server function in the bridge manager (or in some other node, with a pointer to it in the bridge manager).

If broadcast were to be implemented, a likely candidate would be the reservation of a single bus ID, e.g., 0x03FE, to signal a network-wide, unconfirmed broadcast.

Dick Scheel will post an announcement to the relevant reflectors to solicit comments on the necessity of broadcast and how some of these problems might be solved if Serial Bus bridges did not implement a global broadcast.

Multicast IP was also brought up. This might be suitable for running over isochronous channels, since one of the common usages (Mbone) is a fairly steady stream of data. However, there are other usages, such as router table updates, that are very asynchronous in nature.

4.6 Speed capabilities

When packets are in transit across a bus, i.e., from one portal to another for additional forwarding, what speed can be selected to optimize bandwidth utilization? Could an application “ping” the remote node? Perhaps, but it could be time consuming and could be difficult to determine the maximum packet size supported across intervening buses.

Perhaps the bridge manager can determine the lowest speed between any two portals on a Serial Bus and program the portals to use this speed for any packets in transit. Similarly, the final portal in the path (which delivers the packet to the destination node on its local bus) needs to know what speed to use. Again, the bridge manager could at initialization time look up the slowest speed from the portal to any node on that bus, and store that in a single field in the portal. Using these “least common denominator” techniques may not always give the best choice for speed, but it will always give a safe choice. The group felt that in real networks it is likely to be common for all devices on a bus to operate at the same speed. The alternative is to have full routing tables in the portals, with a speed field added to each entry, plus a 63 entry table of speeds to use if the destination is on that portal's local bus.

An entry in the config rom should be defined that specifies whether this portal supports anything more complex than the simple "minimal safe speed" technique. A value of 0 would specify the simple technique, and values 1..n would be reserved for future definition. If a portal indicates that it supports some (yet to be defined) additional capability, and if the bridge manager knows what to do with it, then the bridge manager may set up the additional capability. Even if the portal supports a higher capability, it must also support the simple technique in case the bridge manager only knows how to use that.

Another approach would be to have the bridge manager provide a service: given a 16 bit source address A and a 16 bit destination address B, what is the highest speed and longest packet that may be used in between the two? Actually, the speed would be the initial speed (from the source to the first bridge portal). This could lead to a general capability in the bridge manager to be a server for some set of client requests. The length returned would be based only on the speeds of the links in the path and any buffer space limitations of the bridges in the path. It would be up to the two end nodes to also take into consideration any limitations of the nodes themselves.

The speed/length service might be an optional capability of the bridge manager, and if it isn't implemented then the response to a request for this service would just be hard coded as "S100 speed, S100 max length".

If this server method is used, it is still possible that the values returned could become invalid at a later time during an ongoing conversation between two nodes. This could happen if there is a change in a bus somewhere in the path. The end nodes can see all bus reset notifications, but the end nodes should never need to know what buses are in the path between them. The way this should be handled is to reissue the speed/length request if a transaction to the other node times out (or returns other errors???).

The same generic capability for service in the bridge manager could be utilized to provide solutions to other difficult problems, e.g., allocate isochronous resources to connect talker A on bus X with listeners B and C on buses Y and Z.

5. PAR REVISION

Revision of the PAR was again discussed, since several members needed to leave. It was unanimously approved that Dick Scheel should submit a revised PAR with a completion date of 12/31/97. The group felt that this would be more realistic when allowing time for balloting and other procedural matters.

6. MORE DESIGN DISCUSSIONS

On the second meeting day, the group returned to the topic of speed and packet length when an async packet traverses a network. It was noted that the bus info block in a node's config rom contains a field that specifies the maximum size of an async block write to that node. This should be taken into account by any mechanism that determines a maximum packet size between any two nodes. For example, the Pele chip has a limit of 512 bytes.

Is an async only bridge acceptable? The current draft of the standard says it is.

Dick described the work that had previously been done on isochronous clock distribution and updating time stamps in isochronous packets as they pass through bridges. This still needs to be added to the draft document.

Perhaps there should be a service (in the bridge manager, or possibly some other node) that would set up isochronous paths. A request and response would be defined where a node could specify the talker, listeners, and bandwidth, and the server would try to set up the path. This would include channel and bandwidth allocation on all involved buses, setting plug control registers, etc. The response would indicate success or failure, and perhaps the channel numbers to be used by the talker and each listener.

Dick was assigned an action item to check with IEEE about using their ftp server instead of the one set up at Symbios. This seemed more appropriate for and IEEE activity, now that IEEE seems to have the service operating.

A possible major problem was brought up. So far we have always assumed that only devices built according to the (yet to be completed) 1394.1 specification would ever initiate a remote transaction. Old devices could respond, but never initiate. This has been important in the design of the reset notification / async transaction integrity work. However, we need to look at some of the FCP and AV/C-CTS stuff to see if an old device could be set up by a new device to talk to a remote node. If so, the initiator of a remote transaction might not support the reset notification register. One possible way out of this is that the node that did the setup would see the reset notification and could take steps to stop the old node. We need to look at this.

7. NEXT MEETING DATE

Previously we had set September 17-18 as a meeting date. This has been canceled since we feel we need more development time between meetings. Dick will try to set up a one day meeting on the day before or after the October 1394 Trade Association meeting, since a large majority of P1394.1 committee members will also be attending that meeting. Tentatively the Trade Association meeting will be October 8-10 in Redmond, Washington. If that is the final choice, Dick will try to arrange a P1394.1 meeting on October 7 or 11. He will post the information on the committee reflector.

8. ADJOURNMENT

The meeting was adjourned at approximately noon on 7/30/96.

ATTENDEES:

Herman Lao	3Com	408-764-6124	herman_lao@3mail.3com.com
Waychi Doo	3Com	408-764-6112	waychi_doo@3com.com
Peter Johansson	Congruent Software	510-531-5472	pjohansson@aol.com
Bill Duckwall	Firefly	408-429-9845	p1394@aol.com
Dave Banks	HP Labs / UK	+44 117 922 8244	dmb@hplb.hpl.hp.com
Lou Fasano	IBM Microelectronics	914-892-8904	lfasano@vnet.ibm.com
Jerry Hauck	Intel	408-765-5528	jerry_hauck@ccm11.sc.intel.com
Hugh Curley	Knowledge Tech	303-682-1886	hcurley@aol.com
Katsuya Hasegawa	Matsushita	415-858-1000 x128 katsuya@tadw.research.panasonic.com	
Peter Teng	National Semiconductor	408-721-6594	peter@jedi.nsc.com
Robert Stodieck	Panasonic	408-342-8624	stodieck@masca.com
Calto Wong	Philips	914-945-6382	cxw@philabs.research.philips.com
Gene Milligan	Seagate	405-324-3070	Gene_Milligan@notes.seagate.com
Dick Scheel	Sony	408-955-4295	dicks@lsi.sel.sony.com
Du Hung Hou	Sony	408-955-5742	duh@lsi.sel.sony.com
Mike Winchell	Symbios Logic	470-225-4807	Michael.Winchell@Symbios.COM
Richard Mourn	Symbios Logic	709-533-7421	richard.mourne@symbios.com
Alan Wetzel	TI	214-480-2155	awetzel@ti.com
Burke Henehan	TI	214-480-2603	bhen@msg.ti.com
Henry Angulo	TI	214-480-2117	angu@msg.ti.com